CLAIMS

What is claimed is:

1. A method for managing data, the method comprising the steps of:

maintaining a plurality of persistent data items on persistent storage accessible to a plurality of nodes, the persistent data items including a particular data item stored at a particular location on said persistent storage;

assigning exclusive ownership of each of the persistent data items to one of the nodes, wherein a particular node of said plurality of nodes is assigned exclusive ownership of said particular data item;

when any node wants an operation performed that involves said particular data item, the node that desires the operation to be performed ships the operation to the particular node for the particular node to perform the operation on the particular data item as said particular data item resides at said particular location;

while the first node continues to operate, reassigning ownership of the particular data item from the particular node to another node without moving the particular data item from said particular location on said persistent storage;

after the reassignment, when any node wants an operation performed that involves said particular data item, the node that desires the operation to be performed ships the operation to said other node for the other node to perform the operation on the particular data item as said particular data item resides at said particular location.

2. The method of Claim 1 wherein the step of reassigning ownership of the particular data item from the particular node to another node includes updating an ownership map that is shared among the plurality of nodes.

3. The method of Claim 1 wherein the plurality of nodes are nodes of a multi-node database system.

4. The method of Claim 3 wherein the multi-node database system includes nodes that do not have access to said persistent storage.

5. The method of Claim 3, wherein:

said persistent storage is a first persistent storage of a plurality of persistent storages used by said multi-node database system; and

the method further comprises reassigning ownership of a second data item from a first node that has access to said first persistent storage to a second node that has access to a second persistent storage but does not have access to said first persistent storage; and

wherein the step of reassigning ownership of the second data item includes moving the second data item from said first persistent storage to said second persistent storage.

6. The method of Claim 3 wherein the step of reassigning ownership of the particular data item from the particular node to another node is performed in response to the addition of said other node to said multi-node database system.

7. The method of Claim 3 wherein:

the step of reassigning ownership of the particular data item from the particular node to another node is performed in anticipation of the removal of said particular node from said multi-node database system; and

the method further comprises the step of, in anticipation of the removal of said particular node from said multi-node database system, physically moving from said persistent storage to another persistent storage a second data item that is reassigned from said particular node to a node of said multi-node database system that does not have access to said persistent storage.

8. The method of Claim 3 wherein the step of reassigning ownership of the particular data item from the particular node to another node is performed as part of a gradual transfer of ownership from said particular node to one or more other nodes.

9. The method of Claim 8 wherein the gradual transfer is initiated in response to detecting that said particular node is overworked relative to one or more other nodes in said multi-node database system.

10. The method of Claim 9 wherein the gradual transfer is terminated in response to detecting that said particular node is now longer overworked relative to one or more other nodes of said multi-node database system.

11. The method of Claim 3 wherein the step of reassigning ownership of the particular data item from the particular node to another node is performed as part of a gradual transfer of ownership to said other node by one or more other nodes, wherein said gradual transfer is initiated in response to detecting that said other node is underworked relative to one or more other nodes in said multi-node database system.

12. The method of Claim 3 further comprising the steps of:

after a first node has been removed from the multi-node system, continuing to have a set of data items owned by the first node; and

reassigning ownership of data items from the first node to one or more other nodes in response to detecting requests for operations that involve said data items.

13. The method of Claim 3 further comprising the steps of:

after a first node has been removed from the multi-node system, continuing to have a set of data items owned by the first node; and

reassigning ownership of a data item from the first node to a second node in response to detecting that the workload of said second node has fallen below a predetermined threshold.

14. The method of Claim 1 wherein:

at the time said particular data item is to be reassigned to said other node, the particular node stores a dirty version of said particular data item in volatile memory; and

the step of reassigning ownership of the particular data item from the particular node to another node includes writing said dirty version of said particular data item to said persistent storage.

15. The method of Claim 1 wherein:

at the time said particular data item is to be reassigned to said other node, the particular node stores a dirty version of said particular data item in volatile memory; and the step of reassigning ownership of the particular data item from the particular node to another node includes forcing to persistent storage one or more redo records associated with said dirty version, and purging said dirty version from said volatile memory without writing said dirty version of said particular data item to said persistent storage;

said other node reconstructs said dirty version by applying said one or more redo records to the version of the particular data item that resides on said persistent storage.

16. The method of Claim 1 wherein:

at the time said particular data item is to be reassigned to said other node, the particular node stores a dirty version of said particular data item in volatile memory; and

the method further includes the step of transferring the dirty version of said particular data item from volatile memory associated with said particular node to volatile memory associated with said other node.

17. The method of Claim 16 wherein the step of transferring the dirty version is performed proactively by the particular node without the other node requesting the dirty version.

18. The method of Claim 16 wherein the step of transferring the dirty version is performed by the particular node in response to a request for the dirty version from said other node.

19. The method of Claim 1 wherein:

the step of reassigning ownership of the particular data item from the particular node to another node is performed without waiting for a transaction that is modifying the data item to commit;

the transaction makes a first set of modifications while the particular data item is owned by the particular node; and

the transaction makes a second set of modifications while the particular data item is owned by said other node.

20. The method of Claim 19 further comprising rolling back changes made by said transaction by rolling back the second set of modifications based on undo records in an undo log associated with said other node, and rolling back the first set of modifications based on undo records in an undo log associated with said particular node.

21. The method of Claim 1 wherein method includes the steps of:

the other node receiving a request to update said data item;

determining whether the particular node held exclusive-mode or shared-mode access to the data item;

if the particular node did not hold exclusive-mode or shared-mode access to the data item, then the other node updating the particular data item without waiting for the particular node to flush any dirty version of the data item, or redo for the dirty version, to persistent storage.

22. The method of Claim 1 further comprising the steps of:

in response to transferring ownership of said particular data item to said other node, aborting an in-progress operation that involves said particular data item;

after ownership of the particular data item has been transferred to said particular node, re-executing the in-progress operation.

23. The method of Claim 1 wherein:

an operation that involves said particular data item is in-progress at the time the transfer of ownership of said particular data item is to be performed;

the method further includes the step of determining whether to wait for said in-progress operation to complete based on a set of one or more factors; and

if it is determined to not wait for said in-progress operation to complete, aborting said in-progress operation.

24. The method of Claim 23 wherein said set of one of more factors includes how much work has already been performed by said in-progress operation.

25. A method of managing data, the method comprising the steps of:

maintaining a plurality of persistent data items on persistent storage accessible to a plurality of nodes;

assigning ownership of each of the persistent data items to one of the nodes by

assigning each data item to one of a plurality of buckets; and

assigning each bucket to one of the plurality of nodes;

wherein the node to which a bucket is assigned is established to be owner of all data items assigned to the bucket;

when a first node wants an operation performed that involves a data item owned by a second node, the first node ships the operation to the second node for the second node to perform the operation.

26. The method of Claim 25 wherein the step of assigning each data item to one of a plurality of buckets is performed by applying a hash function to a name associated with each data item.

27. The method of Claim 25 wherein the step of assigning each bucket to one of the plurality of nodes is performed by applying a hash function to an identifier associated with each bucket.

28. The method of Claim 25 wherein the step of assigning each data item to one of a plurality of buckets is performed using range-based partitioning.

29. The method of Claim 25 wherein the step of assigning each bucket to one of the plurality of nodes is performed using range-based partitioning.

30. The method of Claim 25 wherein the step of assigning each data item to one of a plurality of buckets is performed by enumerating individual data-item-to-bucket relationships.

31. The method of Claim 25 wherein the step of assigning each bucket to one of the plurality of nodes is performed by enumerating individual bucket-to-node relationships.

32. The method of Claim 25 wherein the number of buckets is greater than the number of nodes, and the bucket-to-node relationship is a many-to-one relationship.

33. The method of Claim 25 further comprising the step of reassigning from a first node to a second node ownership of all data items that are mapped to a bucket by modifying a bucket-to-node mapping without modifying a data-item-to-bucket mapping.

34. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 1.

35. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 2.

36. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 3.

37. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 4.

38. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 5.

39. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 6.

40. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 7.

41. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 8.

42. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 9.

43. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 10.

44. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 11.

45. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 12.

46. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 13.

47. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 14.

48. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 15.

49. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 16.

50. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 17.

51. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 18.

52. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 19.

53. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 20.

54. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 21.

55. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 22.

56. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 23.

57. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 24.

58. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 25.

59. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 26.

60. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 27.

61. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 28.

62. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 29.

63. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 30.

64. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 31.

65. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 32.

66. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 33.

67. A method for use in a multi-node shared-nothing database system, the method comprising the steps of:

a first node of said multi-node shared-nothing database system initially functioning as exclusive owner of a first data item and a second data item, wherein said first data item and said second data item are persistently stored data items within a database managed by the multi-node shared-nothing database system;

without changing the location of a first data item on persistent storage or shutting down said first node, reassigning ownership of the first data item from the first node to a second node of said multi-node shared-nothing database system; and

after reassigning ownership, the first node continuing to operate as the owner of the second data item, and to handle all requests for operations on said second data item.